Algorithms – Programming Assignment #1 – Spring 2024 – One Step at a Time

A particle is located in an n-dimensional space. Starting from the origin, the particle chooses an axis at random and takes a step of one unit along that axis. The particle is equally likely to choose any axis and it is equally likely to take the step in either direction. (For example, if n=3, the particle is equally likely to choose the x, y, or z axes. Suppose it chose the z axis. Then it is equally likely to move up or down by one unit along the z axis.) After the step is taken, the move is repeated from the new position, and so on.

In this programming assignment, you will consider the following question:

If the particle steps continue forever, what is the probability that the particle returns to the origin if n = 1, 2, and 3?

To address this question, write a program to perform several numerical experiments.

1. Start by writing a function that has a particle in one dimension (n=1) take 20 steps and outputs whether the particle ever returned to the origin. Repeat this experiment 100 times and report the percentage of times the particle returned to the origin.
2. Repeat part 1 for a particle in 2D and 3D.
3. Now repeat parts 1 and 2 for the following numbers of steps: 200; 2,000; 20,000; 200,000; 2,000,000.
4. Once parts 1-3 are working, set up your code to run all of the experiments together and output tables like the ones below with the percentages of times the particle returned to the origin filled in and the overall run time (for 3D only) filled:

```
Percentages of time particle returned to origin:
Number of steps:  20       200      2,000    20,000  200,000   2,000,000
1D
2D
3D

Run time (seconds):
Number of steps:  20       200      2,000    20,000  200,000   2,000,000
3D
```

- Note your work will have values filled in for each value of number of steps and dimension (1D, 2D, and 3D for percentage table; just 3D for the 'Run time' table).
- Your tables need not look exactly like the ones above, but they must have a neat, readable appearance with columns aligned as in the examples above.

Additional Instructions

- Your program must include a main() function that is called. It must also consist of additional functions called by main().
- Follow the PEP8 style as described in the PEP8 class handout.
- A hardcopy **Outline** of your program is due in the D2L Dropbox on 1/26. Follow the guidance from class and the class **Outline** lecture.
- The **final program** is due at the beginning of your class on 2/6. Upload your code and a screenshot of your code run with the results tables to D2L.

** Collaboration Policy **

You are to work **alone** on this assignment.

	Allowed	**Not Allowed**
Collaboration with People	LCDR Neumann	with anyone else
Materials	**your own course materials from this course and from CML**; this means your own notes, your own textbooks, your own class handouts and the D2L sites for these two courses only	any other materials

If you have any questions, ask.